Amendments to the Specification:

Please amend the paragraph beginning at page 2, line 3 as follows:

Figure 1 depicts a block diagram of a conventional system 10 for managing tables of network processors. The system 10 includes a conventional host processor 20 ~~12~~ used by a network administrator and conventional network processors 30, 40, and 50. The conventional host processor 20 ~~10~~ typically includes a table management application 22 that is developed at least in part by the owner of the conventional system 10. The network administrator uses the conventional table management application 22 to configure, update, and otherwise manage the tables, such as lookup tables, of the conventional network processors 30, 40, and 50 in the conventional system 10.

Please amend the paragraph beginning at page 12, line 14 as follows:

If sequential access is supported by a table, then records in a table may also be accessed sequentially from the beginning of the table or from some marked location within the table. Thus, the tables 160 and 170 are depicted as including pointers 161A, 161B, 163A, 163B, 165A, and 165B and 171A, 171B, 173A, 173B, 175A, and 175B. The sequential access to records ~~reords~~ may occur in a forward direction (where the term forward indicates access to the n+1th record following the nth record) or in a backward direction (where the term backward indicates access to the n-1th record following the nth record) or both. If sequential access is supported, then all of the records in the table are circularly linked. Therefore, there is no concept of the beginning (or end) of a table since there is no reference point to support this concept. However, tables may support bookmarks, such as the bookmarks 167, 169, 177, and 179, that may temporarily indicate a reference point for sequential access of records. A bookmark, such as the bookmarks 167, 169, 177, and 179, indicates the starting point of a sequential access. The complete table is said to have been sequentially traversed if the N+1th (or N-1th for a backward access) sequential access returns a bookmarked record.

Please amend the paragraph beginning at page 13, line 18 as follows:

Figure 4C depicts an abstraction of a cascade of tables 182, 184, ~~186, and 188~~ and 186. A table may be cascaded with other tables if the tables support the same type of search key/algorithms. When tables are cascaded, the random search for a particular record extends into the other tables in the cascade until a 'hit' is detected. A search failure is reported only when the search for the requested record fails through all tables in the cascade. Thus, in the cascade 180 a search of the table 182 would ~~wouldu~~ extend into tables 184 and 186. Further, note that the cascade 180 can include both fast tables 182 and 186 as well as a standard table 184. Thus, tables 182, 184, and 186 of different types can be placed in the cascade 180.

Please amend the paragraph beginning at page 28, line 13 as follows:

The Table Unlock ~~Unlcok~~ API is used to moves a table from locked to enabled state. In a preferred embodiment, the only parameter associated with this operation is the Table ID.

Please amend the paragraph beginning at page 34, line 4 as follows:

The Table Start Aging API allows aging of a table to commence. When a table is defined, it is possible to indicate whether aging is supported on the table and if so, the Aging Procedure ID is provided (refer to the Lookup Table Define API for details of the parameter). If aging is supported, the Lookup Table Define API allows the user to be automatically registered for sequential access and returns a Sequential Access ID. However, the aging procedure does not actually start until Table Start Aging API is used. Because the aging procedure involves sequential access, the Sequential Access ID returned as part of the Table Define API operations and is provided as a parameter for this operation. The table for which the Table Start Aging API is utilized ~~utilzied~~ is in the enabled or locked state. The parameters associated with the Table Start Aging API are depicted in Table 12.

Please amend the paragraph beginning at page 35, line 7 as follows:

Figure 5 is high-level flow chart of one embodiment of a method 210 in accordance with the present invention for using a mechanism in accordance with the present invention for managing tables used by network processors. For clarity, the method 210 is described in conjunction with the system 100 depicted in Figure 2. Referring to Figures 2 and 5, the method 210 presumes that the network processors 120, 130, and 140, as well as the table management application 112 have been configured for use with the generic APIs ~~112~~ 150. For example, the table management application 112 is network processor independent and has a generic interface appropriate for use with the generic APIs ~~112~~ 150.